CHAPTER

1

The Essence of UNIX

case ▶ Dominion Consulting specializes in management systems for large hotels and resorts. Dominion's founders, Eli Addison and Carmen Scott, recognize the need for an in-house computer system that lets their employees work as a team. UNIX is an operating system designed for collaborative development of software, allowing people to work together and share information in controlled ways. Dominion has offered you a position as a UNIX system trainee. Your managers want you to understand the basics of operating systems in general, and UNIX in particular. They ask you to log on to UNIX and learn how to use some basic commands.

LESSON A

objectives

After completing this lesson, you should be able to:

- Define operating systems in general and the UNIX operating system in particular
- Describe Linux as it relates to UNIX
- Explain the function of UNIX shells
- Describe options for connecting to a UNIX system
- Define the syntax used for entering UNIX commands
- Use the date, cal, who, man, whatis, and clear commands
- Perform basic command-line editing operations
- Enter multiple commands on a single command line
- Recall a command from the command history
- Log on to and log out of UNIX

Getting Started with UNIX

This chapter introduces the UNIX operating system and a few of its basic commands. It also explains how you can use Linux to learn UNIX. A variant of UNIX, Linux runs on PCs with Intel processors but uses the same file system and commands as UNIX, which usually runs on a network. Using Linux on your PC is virtually the same as using UNIX on a network.

After you explore essential background information in this chapter, you begin to work with UNIX. If you're familiar with operating systems in general, then some background material may be review for you. This chapter also provides plenty of opportunity for hands-on practice of UNIX commands, primarily in the context of the opening case. The case study reflects a realistic scenario for the tasks you complete in this chapter. You learn to use a variety of basic commands to meet the goals of the scenario.

Understanding Operating Systems

An **operating system** (OS) is the most important program that runs on a computer. Operating systems enable you to store information, process raw data, use application software, compile your own programs, and access all hardware attached to a computer, such as a printer or keyboard. In short, the operating system is the most fundamental computer program. It controls all the computer's resources and provides the base upon which application programs can be used or written. Figure 1-1 shows a model of an operating system.

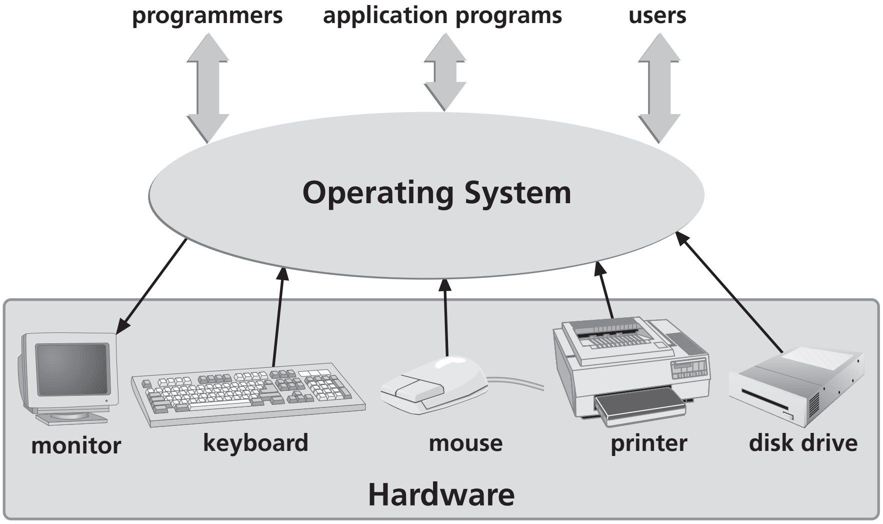


Figure 1-1: Operating system model

Different computer systems may have different operating systems. For example, the most common operating systems for personal computers are DOS, OS/2, and Windows. Mainframe computers may use Digital Equipment's VAX operating system or IBM's System 370. Networks also have operating systems, such as Linux and Windows NT. UNIX is the leading operating system for workstations, which are powerful single-user computers linked together on a local-area network.

PC Operating Systems

A **personal computer** system or **PC** is usually a standalone machine, such as a desktop or laptop computer. A PC operating system conducts all the input, output, processing, and storage operations on a single computer. Figure 1-2 identifies some popular PC operating systems.

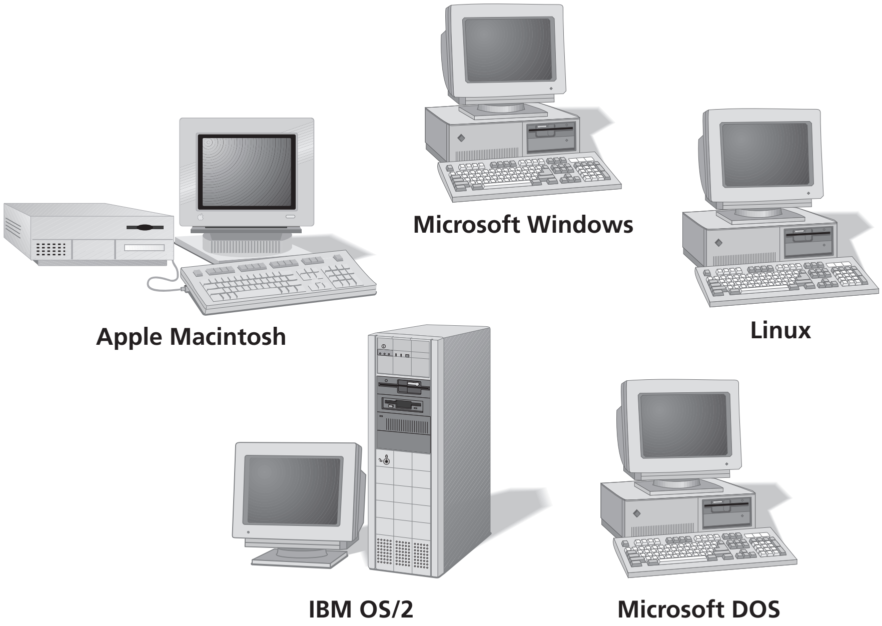


Figure 1-2: Common PC operating systems

Mainframe Operating Systems

A mainframe operating system controls a **mainframe system**, a large computer system with multiple processors that conducts input, output, processing, and storage operations for many users. Historically, mainframe systems have been popular in large corporations and industrial computing. Figure 1-3 shows some recognized mainframe operating systems and their manufacturers.

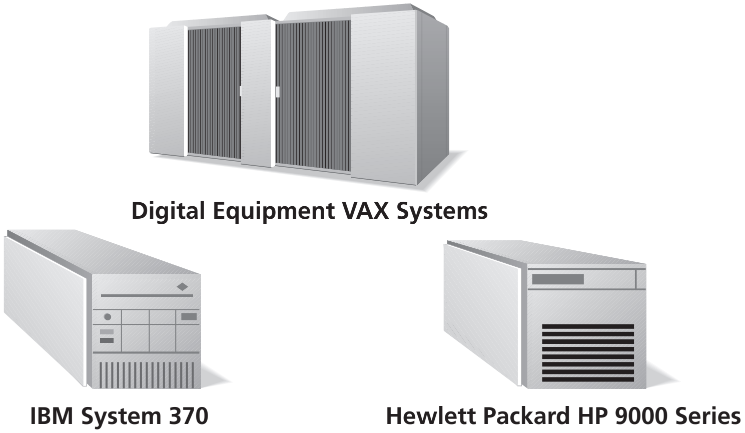


Figure 1-3: Common mainframe operating systems

Network Operating Systems

A computer **network** combines the convenience and familiarity of the personal computer with the processing power of the mainframe. A network lets multiple users share computer resources and files. A **network operating system** controls the operations of a **server** computer, sometimes called a **host** computer, which accepts requests from user programs running on other machines, called **clients**. Figure 1-4 shows the relationship of servers and clients on a network.

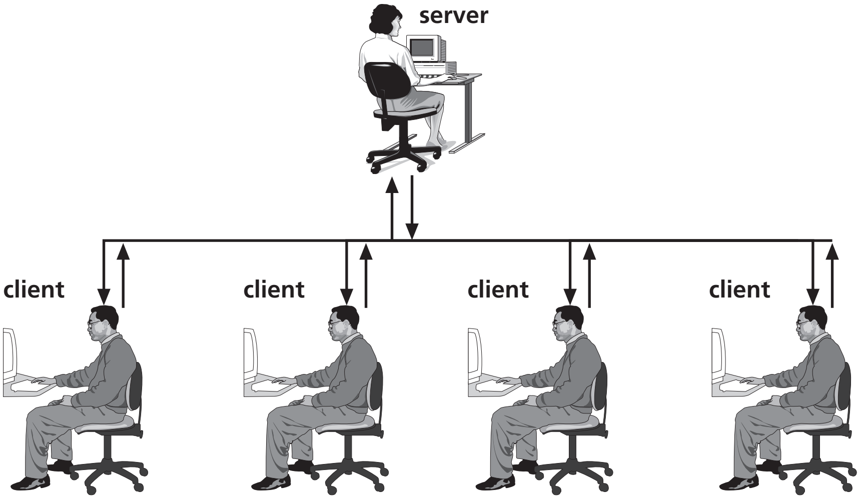


Figure 1-4: Relationship of servers and clients on a network

In a centralized approach, all the users' data and applications reside on the server. This type of network is called a **server-based network**. The system administrator secures all the information on the network by securing the server. The system administrator easily maintains the users' applications and performs back-up operations directly on the server. If the server fails, however, the entire network fails.

Peer-to-peer networks, which work best for small networks, are more distributed than server-based networks. In a peer-to-peer configuration, each system on the network is both a server and a client. Data and applications are not centrally located but reside on the individual systems in the network. Software upgrades and back-up operations must be performed locally at each computer. Security, which is implemented on each computer, is not uniform. Each user of the network is, to some degree, responsible for administering his or her own system. Despite the disadvantages a peer-to-peer network presents to the system administrator, the individual users do not depend on a central server. If one computer in the network fails, the other systems continue to operate.

Introducing the UNIX Operating System

UNIX is a multi-user, multitasking operating system with built-in networking functions. It can be used on systems functioning as:

- Dedicated servers in a server-based network
- Client workstations connected to a server-based network
- Client/server workstations connected to a peer-to-peer network
- Standalone workstations not connected to a network

UNIX is a **multi-user system,** which lets many people simultaneously access and share the resources of a server computer. Users must **log on** by typing their user name and a password before they are allowed to use a multi-user system. This validation procedure protects each user's privacy and safeguards the system against unauthorized use. A **multitasking system** lets one user execute more than one program at a time. For example, on a multitasking system, you can update records in the foreground while your document prints in the background.

UNIX is also a portable operating system. Its **portability** means it can be used in a variety of computing environments. In fact, UNIX runs on a wider variety of computers than any other operating system. It also runs on the Internet, regulating popular programs such as **File Transfer Protocol** (FTP), an Internet protocol used for sending files; and **Telnet**, an Internet terminal emulation program. Dominion Consulting chose UNIX as the OS for its computer system because many of its employees must work on a range of computers performing a variety of tasks at the same time.

A Brief History of UNIX

A group of programmers at Bell Labs originally developed UNIX in the early 1970s. Bell Labs distributed UNIX in its source code form, so anyone who used UNIX could customize it as needed. Attracted by its portability and low cost, universities began to modify the UNIX code to make it work on different machines. Eventually, two standard versions of UNIX evolved: AT&T produced System V and University of California at Berkeley developed BSD. Using features of both versions, Linux may be a more integrated version of UNIX than its predecessors. Currently, the POSIX project, a joint effort of experts from industry, academia, and government, is working to standardize UNIX.

UNIX Concepts

UNIX pioneered concepts that have been applied to other operating systems. For example, Microsoft DOS and Microsoft Windows adopted original UNIX design concepts, such as the idea of a **shell**—an interface between the user and the operating system—and the hierarchical structure of directories and subdirectories.

The concept of layered components that make up an operating system also originated with UNIX. Layers of software surround the computer system's inner core to protect its vital hardware and software components and to insulate the core system and its users. Figure 1-5 shows how the layers of a UNIX system form a pyramid structure.

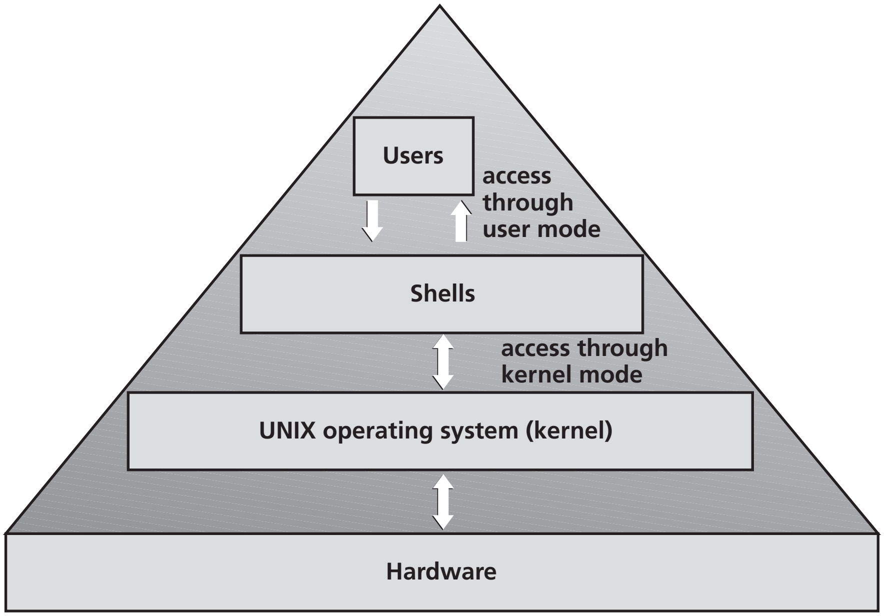


Figure 1-5: Layers of a UNIX system

At the bottom of the pyramid is the hardware. At the top are the users. The layers provide insulation, assuring system security and user privacy. The **kernel** is the base operating system, which interacts directly with the hardware and services the user programs. It is only accessible through **kernel mode**, which is reserved for the system administrator. This prevents unauthorized commands from invading the **foundation layer** or the hardware that supports the entire UNIX structure. **User mode** provides access to higher layers where all application software resides.

This layered approach, and all other UNIX features, were designed by programmers, for programmers to use in complex software development. Because the programmers wrote UNIX in the C programming language, it can be installed on any computer that has a C compiler. Its portability, flexibility, and power make UNIX a logical choice for network operating systems. And, with the emergence of a new PC version called Linux, the popularity of UNIX is increasing.

Linux and UNIX

Linux is a UNIX-like operating system. Linus Torvalds, who released it to the public, free of charge, in 1991, originally created Linux. A number of companies now distribute professional versions of Linux.

Linux offers all the complexity of UNIX at no cost. It is robust enough to handle large tasks with all the networking features of commercial UNIX versions. You can install Linux on your PC, where it can coexist with other operating systems, and test your UNIX skills. All these features make Linux an excellent way to learn UNIX, even when you have access to other computers running on UNIX.

Introducing UNIX Shells

The **shell** is a UNIX program that interprets the commands you enter from the keyboard. UNIX provides several shells, including the Bourne shell, the Korn shell, and the C shell. Steve Bourne at AT&T Bell Laboratories developed the **Bourne shell** as the first UNIX command processor. Another Bell Labs employee, David Korn, developed the Korn shell. Compatible with the Bourne shell, the **Korn shell** includes many extensions, such as a history feature that lets you use a keyboard shortcut to retrieve commands you previously entered. The **C shell** is designed for C programmers' use. Linux uses the freeware **Bash shell** as its default command interpreter. Its name is an acronym for "Bourne Again Shell," and it includes the best features of the Korn and Bourne shells. No matter which shell you use, your communications with UNIX always take place through a shell interpreter. Figure 1-6 shows the role of the shell in the UNIX operating system.

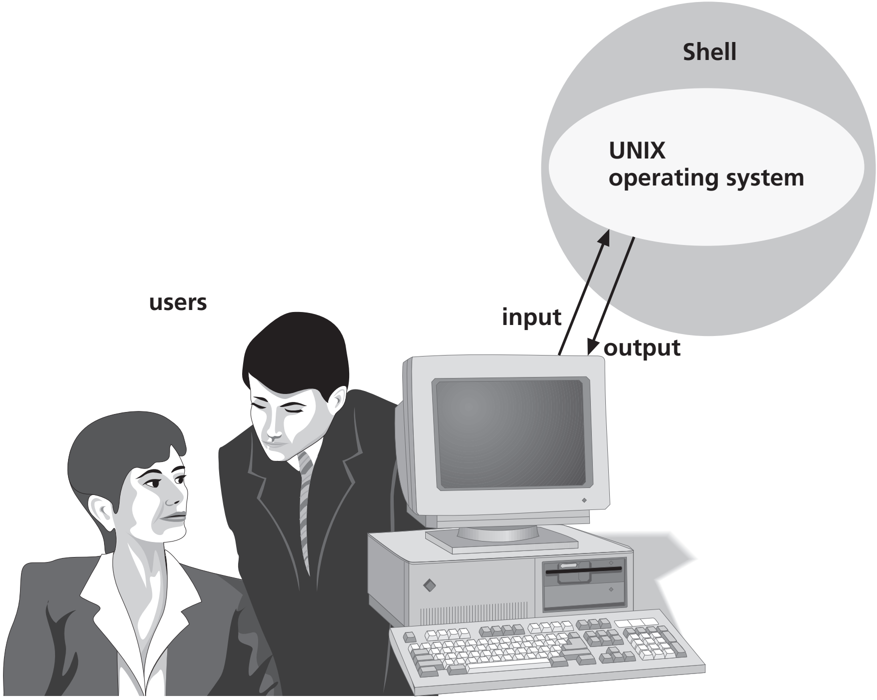


Figure 1-6: Shell's relationship to the user and the hardware

Choosing Your Shell

Before working with a UNIX system, you need to determine which shell will serve as your command interpreter. Shells do much more than interpret commands: each has extensive built-in commands that, in effect, turn them into first-class programming languages. (You will pursue this subject in depth in Chapters 6 and 7.) You choose a shell when the system administrator sets up your user account. Most users choose the Bash shell, although you can choose any of these:

- Bourne
- Korn
- C shell
- Bash
- tcsh (a freeware shell derived from the C shell)
- zsh (a freeware shell derived from the Korn shell)

Switching from Shell to Shell

After you choose your shell, the system administrator stores your choice in your account record, and it becomes your assigned shell. UNIX uses this shell any time you log on. However, you can switch from one shell to another by typing the shell's name (such as **tcsh**, **bash**, or **zsh**) on your command line. You work in that shell until you log on again or type another shell name on the command line. Users often use one shell for writing shell scripts (programs) and another for interacting with a program.

Choosing User Names and Passwords

Before you can work with UNIX and its programs, you must log on by providing a unique user name and password. Decide on a name you want to use to identify yourself to the UNIX system, such as "aquinn." This is the same name others on the UNIX system use to send you electronic mail. UNIX recognizes only the first eight characters of a user name, so choose a user name with eight or fewer characters.

You must also choose a password, which must contain five or more characters. The password should be easy for you to remember but difficult for others to guess, such as your birth date written in a mix of uppercase and lowercase letters, numbers, and hyphens. The password can contain letters, numbers, and punctuation symbols but not control characters.

You can log on to any UNIX or Linux system as long as you have a user account and password on the host (server) computer. A UNIX system administrator creates your account by adding your user name (also called a *login name* or *user id*) and your password. To use this book and the hands-on tutorials, you must connect to and establish an account on a UNIX or Linux system using one of these methods:

- Through a Telnet connection to a remote computer
- As a client on a UNIX client/server network
- As a peer on a peer-to-peer local-area network in which each computer has the Linux operating system installed

- On a standalone PC that has the Linux operating system installed
- Through a log-on terminal, such as a Wyse terminal, connected to a communication port on a UNIX host

The steps you take to connect to a UNIX system vary according to the kind of connection you use. The simplest connection is that of a user terminal that presents the log-on prompt as soon as you turn it on. You can also access the Internet through an Internet Service Provider (ISP) and connect to a remote UNIX host.

Connecting to UNIX Using Telnet

Telnet is a terminal emulation program for the Internet. It runs on your computer and connects your PC to a server, or host, on the network. You can then log on to a UNIX host and begin working with UNIX.

Each computer on the Internet has an **Internet Protocol** (IP) address. An IP address is a set of four numbers separated by periods, such as *172.16.1.61*. Most systems on the Internet also have a **domain name** such as *Lunar.campus.edu*. Both the IP address and the domain name identify a system on the network. Programs such as Telnet use IP addresses or domain names to access remote systems.

To access a UNIX host via Telnet:

1 Find the remote host's IP address or domain name.

2 Connect to the Internet, if necessary. If you use a PPP connection to dial into an ISP, make that connection. If you use a full-time Internet connection at work or school, ignore this step.

3 Start your Telnet program and connect to the UNIX system.

Follow the instructions in your Telnet program to connect to a remote host. Usually, you must provide the host name to connect to a UNIX system. For example, you can type the following command after the prompt in a Windows 95/98 MS-DOS window to gain access to the system *Lunar.campus.edu*.

```
telnet Lunar.campus.edu
```

Logging On to UNIX

After you connect to a UNIX system, you must log on by entering your user name and password. You see a prompt requesting the login or user name and the password. For security reasons, the password does not appear on the screen as you type it. You cannot log on without an authorized user account, if your password fails, or if you wait too long before entering your name and password. Contact your system administrator for help.

After you log on, you are ready to begin using the system.

Entering Commands

To interact with UNIX, you enter a **command**, text you type after the command prompt. When you finish typing the command, press Enter. UNIX is **case-sensitive**, that is, it distinguishes between uppercase and lowercase letters, so that *John* differs from *john*. You type most UNIX commands in lowercase. Commands are divided into two categories: user-level commands that you type to perform tasks, such as retrieve information or communicate with other users; and system-administration commands, which the system administrator uses to manage the system.

You must know a command's syntax to enter it properly. **Syntax** refers to a command's format and wording, as well as the options and arguments you can use to extend and modify its functions. Most commands are single words, such as *clear*. If you enter a command using correct syntax, UNIX executes the command. Otherwise, you receive a message that UNIX cannot interpret your command.

tip

Appendix A, "Syntax Guide to UNIX Commands," alphabetically lists all the commands in this book and tells you how to enter each command and use its options.

The place on the screen where you type the command is called the **command line**. Commands use the following syntax:

Syntax **command_name** [-options] [arguments]

Dissection

- The **command_name** specifies what operation to perform. In the syntax illustrations in this book, command names appear in boldface.
- Command **options** are ways to request that UNIX carry out a command in a specific style, or variation. Options follow command names, separated by a space. They always begin with a hyphen (-). Options are also case-sensitive. For example, **-R** differs from **–r**. You do not need to type an option after every command; however, some commands do not work unless you specify an option. The syntax illustrations in this book list options in square brackets ([ ]) when the command does not require them.
- Command **arguments** follow command options, separated by a space. Command arguments are usually file and directory names. In the syntax illustrations in this book, arguments appear in italics. Square brackets surround arguments if the command does not require them.

The date Command

Use the UNIX date command to display the system date, which the system administrator maintains. Because the date and time on a multi-user system are critical for smooth processing, only the system administrator can change the date. You start your UNIX training at Dominion Consulting by checking the system date.

To display your system date:

1 Type **date** in the command line, and press **Enter**.

A date like the one below appears:

```
Mon Apr 20 21:30:08 EST 2000
```

You may see the abbreviation EDT (eastern daylight time) instead of EST (eastern standard time), or another time zone abbreviation, such as PDT (Pacific daylight time) or CST (central standard time). Notice also that UNIX uses a 24-hour clock.

2 Type **Date** in the command line, and press **Enter**. You see the following system error message:

```
bash: Date: command not found
```

The system error message appears because you must enter the date command, like most UNIX commands, in lowercase letters.

The date command has an option, -u, which displays the time in Greenwich mean time.

To display your system date:

■ Type **date -u** in the command line, and press **Enter**.

A date like the one below appears:

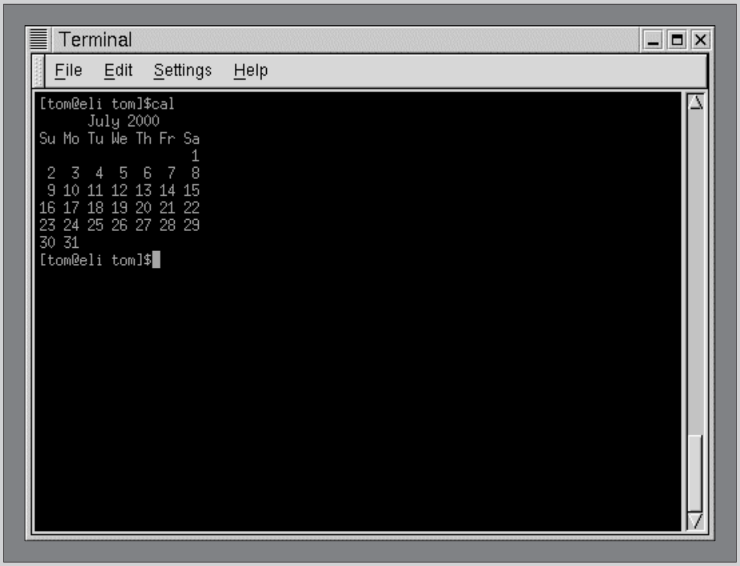
```
Mon Apr 20 23:43:148 UTC 2000
```

The cal Command

Use the cal command to show the system calendar. Your manager at Dominion, Rolfe Williams, advises you that this is the command commonly used to schedule tasks and events.

To display your system's default calendar:

1 Type **cal** in the command line, and press **Enter**. Without an option, the cal command shows a calendar of the current month. Assuming the current month is July of the year 2000, you see the default calendar shown in Figure 1-7.

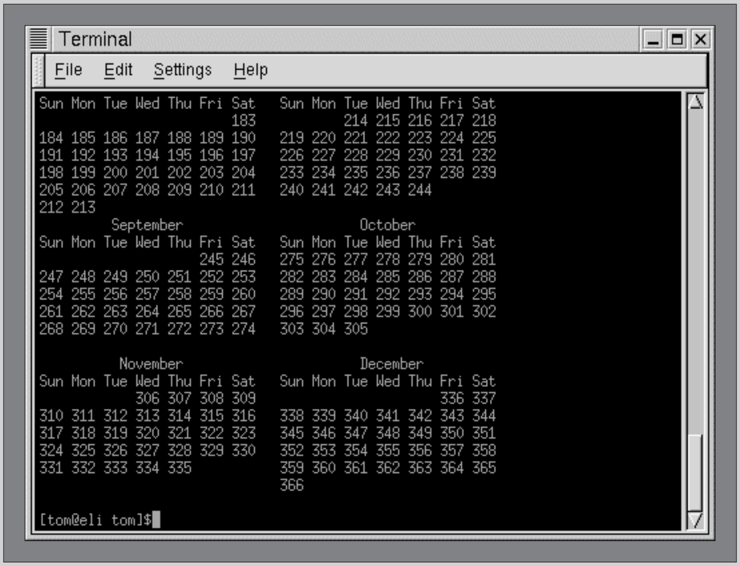


```
[tom@eli tom]$cal
      July 2000
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
[tom@eli tom]$
```

Figure1-7: Example of cal command (current month)

2 Type **cal –j 2000** in the command line, and press **Enter**.

The –j option displays the Julian date format. In other words, it shows the days as numbers starting with 1 and ending with 366, as shown in Figure 1-8.



```
Sun Mon Tue Wed Thu Fri Sat   Sun Mon Tue Wed Thu Fri Sat
                        183           214 215 216 217 218
184 185 186 187 188 189 190   219 220 221 222 223 224 225
191 192 193 194 195 196 197   226 227 228 229 230 231 232
198 199 200 201 202 203 204   233 234 235 236 237 238 239
205 206 207 208 209 210 211   240 241 242 243 244
212 213
         September                      October
Sun Mon Tue Wed Thu Fri Sat   Sun Mon Tue Wed Thu Fri Sat
                    245 246   275 276 277 278 279 280 281
247 248 249 250 251 252 253   282 283 284 285 286 287 288
254 255 256 257 258 259 260   289 290 291 292 293 294 295
261 262 263 264 265 266 267   296 297 298 299 300 301 302
268 269 270 271 272 273 274   303 304 305

          November                      December
Sun Mon Tue Wed Thu Fri Sat   Sun Mon Tue Wed Thu Fri Sat
            306 307 308 309                       336 337
310 311 312 313 314 315 316   338 339 340 341 342 343 344
317 318 319 320 321 322 323   345 346 347 348 349 350 351
324 325 326 327 328 329 330   352 353 354 355 356 357 358
331 332 333 334 335           359 360 361 362 363 364 365
                              366

[tom@eli tom]$
```

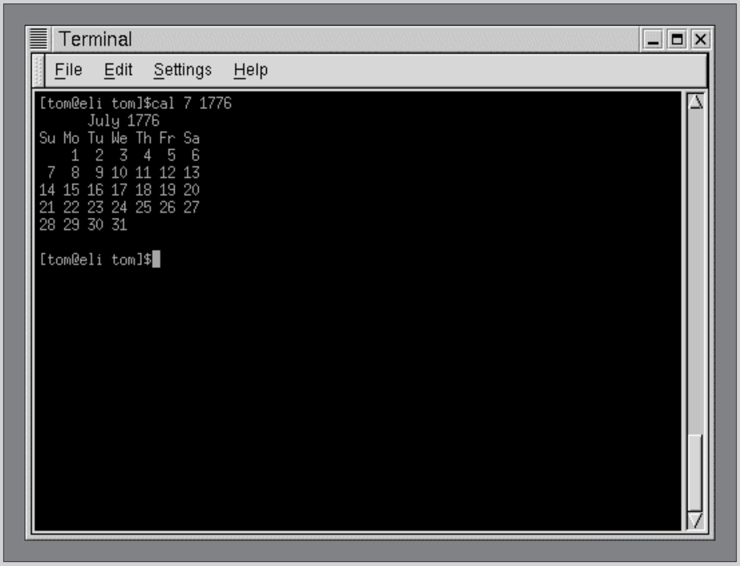
Figure 1-8: Example of cal command (full year—Julian dates)

3 To determine the day of the week when the Declaration of Independence was signed, type **cal 7 1776** in the command line, and press **Enter**. You should see a calendar similar to the one in Figure 1-9. In this case, the month and year are the command arguments.

help

If you type **cal may 1999**, you see an error message because you must use numbers to indicate months, such as *5* for *May*.

```
cal: illegal month value: use 1-12
```



```
[tom@eli tom]$cal 7 1776
      July 1776
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

[tom@eli tom]$
```

Figure 1-9: Example of cal command (July 1776)

The who Command

In a multi-user system, knowing who is logged on to the system may be helpful. Use the who command to see who is using the system and their current location. At Dominion, who is a useful command for getting to know your co-workers.

To use the who command to determine who is logged onto the system:

1 Type **who** in the command line, and press **Enter**.

2 You see a list like the one below showing user names, the terminals they are using, and the dates and the times they logged on.

```
root    tty1   Aug 12 07:56
ellen   tty1   Aug 12 08:15
```

```
john   tty2   Aug 12 08:15
jerry  tty3   Aug 12 08:21
```

3 To display a line of column headings with the who command's output, type **who –H** and press **Enter**. You see a list like the one below.

```
USER   LINE   LOGIN-TIME    FROM
root   tty1   Aug 12 07:56
ellen  tty1   Aug 12 08:15
john   tty2   Aug 12 08:15
jerry  tty3   Aug 12 08:21
```

If any current users are logged on from a remote host, the FROM column shows the name of the host.

4 Idle time is the amount of time that has elapsed with no activity in a user's session. Type **who –i** and press **Enter** to see each user's idle time. You see a list similar to the one below.

```
root   tty1   Aug 12 07:56  00:29
ellen  tty1   Aug 12 08:15    .
john   tty2   Aug 12 08:15  00:01
jerry  tty3   Aug 12 08:21    old
```

The output shows that the person logged on as *root* has performed no activity in the last 29 minutes, and John's session has been idle for one minute. The period on Ellen's line indicates that her session has been active in the last minute. The word *old* on Jerry's line indicates no activity in the past 24 hours.

5 If you wish to use multiple options on the same command line, type them all after a single hyphen. For example, type **who –iH** and press **Enter** to see a list of users with idle times and column headings. You see a list similar to the one below.

```
USER   LINE   LOGIN-TIME    IDLE    FROM
root   tty1   Aug 12 07:56  00:29
ellen  tty1   Aug 12 08:15    .
john   tty2   Aug 12 08:15  00:01
jerry  tty3   Aug 12 08:21    old
```

6 Type **who –q** and press **Enter** to see a quick list of current users. You see a list similar to the one below, which shows only login names and the total number of users on the system.

```
root ellen john jerry
# users=4
```

7 To determine which terminal you are using or what time you logged on, type **who am I** in the command line, and press **Enter**. You see a line similar to the one below. It shows your user name, terminal, and the date and time you logged on.

```
lucky.campus.edu!ellen   tty1   Aug 12 08:15
```

The output above shows that you are logged on to the system lucky.campus.edu as the user ellen.

Note: Any time you provide two arguments to the who command, you'll see the output described in Step 7. For example, you can type who are you, or who x x, to see the same information. Traditionally, UNIX users type who am I to see information describing their session.

Command-line Editing

Shells support certain keystrokes for performing command-line editing. For example, Bash (which is the default Linux shell) supports the left and right arrow keys, which move the cursor on the command line. Other keys, used in combination with the Ctrl or Alt key, cause other editing operations.

Note: Not all shells support command-line editing in the same manner. The following steps work with the Bash shell.

To edit a command typed on the command line:

1 To determine which shell you are using, type **echo $SHELL** and press **Enter**. If you are using the Bash shell, you will see the following output.

```
/bin/bash
```

If you are not using the Bash shell, type **bash** and press **Enter**.

2 Type **who am I**, but do *not* press **Enter**.

3 Press the **left arrow** key to move the cursor to the letter **a** in the word *am*.

4 Press **Alt+D** to delete the word "am."

5 Press **Ctrl+K** to delete the command line from the current cursor position.

6 Press **Ctrl+A** to move the cursor to the beginning of the command line.

7 Press **Ctrl+K** again to delete the command line.

8 Retype the command **who am I**, but do *not* press **Enter**.

9 Press **Alt+B** three times. Watch the cursor move to the first character of the previous word each time you press the key combination. The cursor should be positioned at the beginning of the line.

10 Press **Alt+F** three times. Each time you press the key combination, the cursor moves to the position just before the first character of the next word.

11 Press **Ctrl+A**, and then press **Ctrl+K** to clear the command line.

Multiple Command Entry

You may type more than one command on the command line by separating each command with a semicolon (;). When you press Enter, UNIX executes the commands in the order you entered them.

To enter multiple commands on the command line:

- Type **date ; who –iH** and press **Enter**. You see information similar to the following:

  ```
  Mon Apr 20 21:35:09 EST 2000
  USER   LINE   LOGIN-TIME    IDLE    FROM
  root   tty1   Apr 20 07:56  00:29
  ellen  tty1   Apr 20 08:15    .
  john   tty2   Apr 20 08:15  00:01
  jerry  tty3   Apr 18 08:21    old
  ```

  The date command produces the first line of the output shown above. The remainder of the output is the result of the who command.

The clear Command

As you continue to enter commands, your screen may become cluttered. Unless you need to refer to commands you previously entered and their output, you can use the **clear** command to clear your screen. It has no options or arguments.

To clear the screen:

- Type **clear** on the command line, and press **Enter**. The command prompt is now in the upper-left corner of your screen.

The Command-line History

Often you find yourself entering the same command several times within a short period of time. Most shells keep a list of your recently used commands and allow you to recall a command without retyping it. You can access the command history with the up and down arrow keys. Pressing the up arrow key once recalls the most recently used command. Pressing the up arrow key twice recalls the second most recently used command. Each time you press the up arrow key, you recall an older command. Each time you press the down arrow key, you scroll forward in the command history. When you locate the command you want to execute, press Enter.

To use the command-line history:

1. Type **date** and press **Enter**.
2. Type **who** and press **Enter**.
3. Type **who –iH** and press **Enter**.
4. Type **clear** and press **Enter**.
5. Press the **up arrow** key four times. The date command is recalled to the command line. Do *not* press **Enter**.
6. Press the **down arrow** key twice. The who –iH command is recalled to the command line. Press **Enter** to execute the command.

The man Program

For reference, UNIX includes an online manual that contains all commands, including their options and arguments. The man program in UNIX displays this online manual, called the man pages, for command-line-assistance. Although the man pages for some commands contain more information than others, most man pages list the following items.

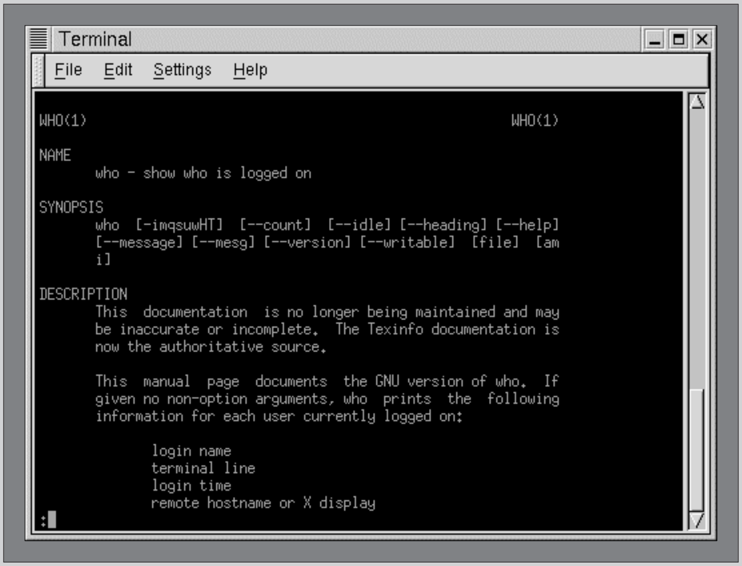
- Name: the name of the command and a short statement describing its purpose
- Synopsis: a syntax diagram showing the usage of the command
- Description: a more detailed description of the command than the name item gives
- Options: a list of command options and their purposes
- See Also: other commands or man pages that provide related information
- Bugs: a list of the command's known bugs

The man program usually accepts only one argument—the name of the command about which you want more information. The online manual shows the valid command formats that your system accepts. To close the online manual, type q.

You decide to display information about the who command and the man program for yourself and others at Dominion.

To display online help for commands:

1. Type **man who** in the command line, and press **Enter**. You see the explanation of the who command illustrated in Figure 1-10.



```
WHO(1)                                                          WHO(1)

NAME
       who - show who is logged on

SYNOPSIS
       who  [-imqsuwHT]  [--count]  [--idle] [--heading] [--help]
       [--message] [--mesg] [--version] [--writable]  [file]  [am
       i]

DESCRIPTION
       This  documentation  is no longer being maintained and may
       be inaccurate or incomplete.  The Texinfo documentation is
       now the authoritative source.

       This  manual  page  documents  the GNU version of who.  If
       given no non-option arguments, who  prints  the  following
       information for each user currently logged on:

              login name
              terminal line
              login time
              remote hostname or X display
:
```

Figure 1-10: Manual page

2 Type **q** to exit the man program.

3 Type **man man** and press **Enter**. You see the man pages describing the man command.

4 Type **q** to exit the man program.

The whatis Command

Sometimes you find that the man pages contain more information than you want to see. To display a brief summary of a command, use the whatis command. The whatis command shows only the name and brief description that appears near the top of a command's man page.

To display a brief description of a command with the whatis command:

1 Type **whatis who** and press **Enter**.

2 You see a summary of the who command, as shown below.

```
who (1)           - show who is logged on
```

3 Type **q** to exit the man program.

Note: The whatis command relies on information stored in a database. The system administrator must execute the makewhatis command, which creates the database, before the whatis command will operate properly. If the whatis command does not display command summaries on your system, the system administrator probably has not executed the makewhatis command.

Logging Out of UNIX

When you finish your day's work or leave your terminal for any reason, log out of the UNIX system. **Logging out** ends your current process and indicates to UNIX that you are finished. How you log out depends on the shell you are using. For the Bourne, Korn, or Bash shells, type exit on the command line or press Ctrl+D. In the C shell, type logout on the command line.

To log out of UNIX:

- In the command line, type **exit** and press **Enter**.

LESSON B

objectives

After completing this lesson, you should be able to:

- **Discuss the role of the system administrator**
- **Identify the system administrator's and the ordinary user's command prompts**
- **Change your personal password**
- **View files on your screen using the cat, more, less, head, and tail commands**
- **Redirect output to a file**

Roles of the System Administrator and Ordinary Users

Understanding the Role of the UNIX System Administrator

There are two types of users on a UNIX system: the system administrator and ordinary users. As the name suggests, the **system administrator** manages the system by adding new users, deleting old accounts, and ensuring that the system performs services well and efficiently for all users. **Ordinary users** are all other users. The system administrator is also called the **superuser**, because the system administrator has unlimited permission to alter the system. UNIX grants this permission when the operating system is initially installed. The system administrator grants privileges and permissions to regular users.

The system administrator has a unique user name: **root**. The system administrator **owns** the root account, which means no one else can be assigned to that account. The password for the root account is confidential; only the system administrator and a back-up person know it. If the root's password is lost or forgotten, the system administrator uses an emergency rescue procedure to reset the password.

The System Administrator's Command Line

While ordinary users type their commands after the $ (dollar sign) command prompt, the system administrator's prompt is the # (pound) symbol. The UNIX system generates a default setting for the command prompt for the system administrator in the following format:

```
[root@hostname]#
```

In the prompt, *hostname* is the name of the computer the system administrator logged on to.

For more information on the role of the system administrator, see Chapter 11, "UNIX/Linux System Administration."

The Ordinary User's Command Line

The $ (dollar sign) is traditionally associated with ordinary users. The UNIX system generates a default setting for the command prompt for ordinary users in the following format:

```
[user name@hostname]$
```

In the prompt, *user name* is the user's login name, such as *jean*, and *hostname* is the name of the computer the user is logged on to.

Changing Passwords

Your user name, or login name, identifies you to the system. You may choose your own user name and give it to the system administrator, who then adds you as a new user. As mentioned before, UNIX recognizes up to eight characters in your user name, which is often your first name or nickname. The user name is unique, but not confidential, and may be provided to other users. The password, on the other hand, is confidential and secures your work on the system. You can change your password, if necessary, by using the passwd command. If you do not have a password, use the passwd command to create one.

UNIX lets you change your password only if the new one differs from the old password by at least three characters; the password has more than five characters, including at least two letters and one number; and the password is different from your user name.

The password you chose for the Dominion system—Gscott956—is too similar to another password on the system, so the system administrator asks you to change it.

To change your password:

1. Type **passwd** after the command prompt, and press **Enter**.
2. Type your new password and press **Enter**. Your new password does not appear on the screen as you type.
3. Retype your new password and press **Enter** so that UNIX can confirm the new password.

help

If the password you retype as confirmation does not match your new UNIX password, UNIX asks you to enter the password again. UNIX may also ask you to choose a different password because you chose one that is too short or too easily guessed, such as *password*.

After changing your password, you should log out and log on again to make sure UNIX recognizes your new password.

tip

Write down your password and store it in a secure place. You need your password everytime you log on to UNIX.

Viewing Files Using the cat, more, less, head, and tail Commands

Three UNIX commands let you view the contents of files: cat, more, and less. The more and less commands display a file, one screenful at a time, while the cat command displays the whole file at once. Two other commands, head and tail, let you view the first few or last few lines of a file.

The cat command gets its name from the word **concatenate**, which means to link. You can display multiple files by entering their file names after the cat command and separating each with a space. UNIX then displays the file's contents in the order you entered them.

Your system administrator at Dominion mentions that you can use the cat command to view a file called shells that resides in the /etc directory. This file contains a list of valid shell programs on the system.

To view the shells file:

1. Type **cat /etc/shells** after the command prompt, and press **Enter**.

   Use the forward slash (/) to indicate a directory or folder change. You see a list of the available shells, including /bin/bash, /bin/sh, /bin/bsh, /bin/tcsh, /bin/csh, and /bin/zsh.

   Sometimes it is helpful to see a file's contents displayed with line numbers. The –n option causes the cat command to display a number at the beginning of each line of output.

2. Type **cat –n /etc/shells** and press **Enter**. You see the same list of shells as before, but this time a number precedes each line.

You can also view another file in the /etc directory called termcap. This multiple-page file contains many specifications about all terminals supported on the Linux system. The cat command is not a practical way to view this file, which is longer than one screen (23 lines). However, you can use the more and less commands to read a large file screen by screen.

To view the contents of large files on the screen with the more command:

1. Type **more /etc/termcap** after the command prompt, and press **Enter**.
2. Press the **spacebar** to scroll to the next screen. You cannot use the more command to return to a previously displayed screen.
3. Terminate the display by typing **q** (for quit).

To view the contents of large files on the screen with the less command:

1. Type **less /etc/termcap** after the command prompt, and press **Enter**. You see a long file of text on your screen.
2. Press the **down arrow** key several times to scroll forward in the file one line at a time.
3. Press the **up arrow** key several times to scroll backward in the file one line at a time.
4. Press **Pg Dn**, **Space**, **z**, or **f** to scroll forward one screen.
5. Press **Pg Up** or **b** to return to a previous screen.
6. Terminate the display by typing **q** (for quit) when you see a colon (:) at the bottom of a screen.

Sometimes you only need to glimpse part of a file's contents to determine what is stored in the file. The head command shows you the first few lines of a file—by default, the first 10 lines.

To view the first or last few lines of a file:

1. Type **head /etc/termcap** and press **Enter** to see the first 10 lines of the /etc/termcap file.
2. The –n option specifies the number of lines the head command displays. Type **head –n 5 /etc/termcap** and press **Enter**. You see the first five lines of the /etc/termcap file.
3. The tail command shows you the last few lines of a file. Like the head command, tail displays 10 lines by default. Type **tail /etc/termcap** and press **Enter** to see the last 10 lines of the /etc/termcap file.
4. The –n option specifies the number of lines the tail command displays. Type **tail –n 5 /etc/termcap** and press **Enter**. You see the last five lines of the /etc/termcap file.

Redirecting Output

In UNIX, the greater than sign (>) is called the redirection symbol. You can use the **redirection symbol** to create a new file or overwrite an existing file by attaching it to a command that produces output. In effect, you "redirect" the output to a disk file instead of the monitor. You already used the who command to find out who was logged on to the system at Dominion. Now you can use the same command with the redirection symbol to save this information in a text file.

To save to a file that lists persons logged on to the system:

1. Type **who > current_users** after the command prompt, and press **Enter**. The who command output does not appear on the screen but is redirected to a new disk file called current_users. UNIX places this text file in the active directory (the folder on the disk where you are currently using the system).
2. Type **cat current_users** after the command prompt, and press **Enter** to see a list of users currently using the system, similar to the one below.

```
jean     tty1      Feb 7  07:15
joseph   tty2      Feb 7  07:15
becky    tty3      Feb 7  08:05
```

You can also use the redirection symbol with the cal command to save a calendar in a text file. For example, you will be involved in a Dominion development project with a projected deadline in the year 2000. You can save the calendar in a text file.

To save the year 2000 calendar in a file:

1. Type **cal 2000 > year_2000** after the command prompt, and press **Enter**. This creates a text file called year_2000.
2. Type **less year_2000** and press **Enter** to see the calendar created by the previous command. Use the arrow keys, Pg Dn, Pg Up, and other keys to scroll through the file.
3. Terminate the display by typing **q** (for quit).

You can also use the cat command to create files from information you type at the keyboard. Type cat>*filename* after the command prompt, where *filename* is the name of the file you are creating. Then press Ctrl+D to end data entry from the keyboard.

Use the redirection symbol (>) to send output to a file that already exists only if you want to overwrite the current file. To append output to an existing file, use two redirection symbols (>>). This adds information to the end of an existing file without overwriting that file.

As you work with UNIX, you remember that Rolfe Williams, your supervisor, asked you to complete a few tasks by the end of the week. You decide to create a notes file of task reminders.

To create a new file:

1. Type **cat > notes** after the command prompt, and press **Enter**.
2. Type the following: **Remember to order a new CD-ROM and send the report by Thursday**, and press **Enter**.
3. Press **Ctrl+D**.

4 To review the file you just created, type **cat notes** after the command prompt and press **Enter**. The sentence you typed in Step 2 appears on the screen.

After you create the notes file, you remember that Rolfe Williams asked you to complete another task. You can append the reminder to the existing notes file. You also want to include the appropriate monthly calendar in the file for reference.

To add information to an existing file:

1. Type **cat >> notes** after the command prompt, and press **Enter**.
2. Type the following: **Also remember to make reservations for Sept. conference**, and press **Enter**.
3. Press **Ctrl+D**.
4. To add the calendar to your notes, type **cal 9 2000 >> notes**.
5. Type **less notes** to review the file.

SUMMARY

- The operating system is the most fundamental computer program. It controls all computer resources and provides the base upon which application programs can be used or written.
- A server-based network is centralized. All the users' data and applications reside on the server, which is secured, maintained, and backed up by the system administrator. Each computer in a server-based network relies on the server. All systems in a peer-to-peer network function as both server and client. The security and maintenance of the network is distributed to each system. If one of the systems in a peer-to-peer network fails, the other systems continue to function.
- The UNIX operating system is a multi-user system that lets many people access and share the computer simultaneously. It is also a multitasking operating system: it can perform more than one task at one time.
- UNIX systems may be configured as dedicated servers in a server-based network, client workstations in a server-based network, client/server workstations in a peer-to-peer network, or standalone workstations connected to no network.
- The concept of the layered components that make up an operating system originated with UNIX. Layers of software surrounding the computer system's inner core protect the vital hardware and software components and insulate the core system from users.
- Linux is a UNIX-like operating system that you install on your PC. It coexists with other operating systems such as Windows and MS-DOS.

- In UNIX, you communicate with the operating system programs through an interpreter called the *shell*, which interprets the commands you enter from the keyboard. UNIX provides several shell programs, including the Bourne, Korn, and C shells. The Bash shell provides enhanced features from the Bourne and the Korn shells. It is the most popular shell on the Linux system.
- In UNIX, the system administrator sets up accounts for ordinary users. To set up your account and to protect the privacy and security of the system, you select and give the system administrator your user name and password. You can log on to any UNIX or Linux system anywhere as long as you have a user account and password on the host (server) computer. You can also use the Microsoft Windows Telnet program to log on to a remote UNIX system.
- The commands you type to work with UNIX have a strict syntax that you can learn by referring to the online manual called the *man pages*. Use the man program to display the syntax rules for a command. Use the whatis command to see a brief description of a command. Use the who command to list who is logged in and where they are located. Use the cal command to display the system calendar for all or selected months. To log out when you decide to stop using UNIX, use the exit or logout command.
- Most shells provide basic command-line editing capabilities and keep a history of your most recently used commands. Use the up and down arrow keys to scroll backward and forward through the list of recently used commands. You may enter multiple commands on a single command line by separating them with a semicolon. UNIX executes the commands in the order you enter them.
- You can use the view commands to view the contents of files. Use the cat command to create a file by typing information from the keyboard. Use the less and more commands to display multi-page documents. Use the head and tail commands to view the first or last few lines of a file.

COMMAND SUMMARY

Chapter 1 commands

Command	Purpose	Options covered in this chapter
cal	Show the system calendar	-j displays the Julian date format
cat	Display multiple files	-n displays line numbers

Chapter 1 commands (continued)		
clear	Clear the screen	
date	Display the system date	-u displays the time in Greenwich mean time
exit or logout	Exit UNIX	
head	Display the first few lines of a file	-n displays the first *n* lines of the specified file
less	Display a long file one screen at a time and scroll up and down	
man	Display the online manual for the specified command	
more	Display a long file one screen at a time and scroll down	
passwd	Change your UNIX password	
tail	Display the last few lines of a file	-n displays the last *n* lines of the specified file
whatis	Display a brief description of a command	
who	See who is logged on to UNIX	-H displays column headings -i displays session idle times q displays a quick list of users

REVIEW QUESTIONS

1. UNIX commands are case-sensitive. In which case must you type them?
a. all uppercase
b. all lowercase
c. sentence case, so the first letter is capitalized and others are lowercase
d. any combination of uppercase and lowercase

2. In which order are UNIX commands given?
a. name, arguments, options
b. name, options, arguments
c. options, name, arguments
d. arguments, options, name

3. Which of these statements is false?
   a. Linux is compatible with UNIX.
   b. Ordinary users cannot change the system date.
   c. The UNIX operating system is called the kernel.
   d. You cannot change the shell after you select one.

4. In a ________________ network, all users' data and applications are centrally located on one system.
   a. peer-to-peer
   b. server-based
   c. standalone
   d. workstation

5. UNIX may be configured to run as ________________.
   a. a server in a server-based network
   b. a client workstation in a server-based network
   c. a client/server workstation in a peer-to-peer network
   d. all of the above

6. What is required to log on?
   a. a user name and password
   b. only a password
   c. a kernel
   d. a shell

7. Use the UNIX date command to ________________.
   a. display the system calendar
   b. display any date in the future
   c. reset the system date
   d. display the system date

8. The man program provides ________________.
   a. a programming guide
   b. an online reference for command syntax
   c. a file template
   d. an operating system version

9. The whatis command provides ________________.
   a. a brief description of a command
   b. a more detailed description of a command than that given by the man program
   c. a description of a hardware device attached to the system
   d. an online tutorial on the UNIX operating system

10. By default, the cal command displays ________________.
    a. the current system date
    b. a list of shells
    c. the current month and year
    d. the contents of the cal file

11. What type of operating system best describes UNIX?
a. a multi-user, multitasking operating system
b. an operating system that conducts all its operations on a standalone PC
c. an operating system with a graphical user interface
d. a portable single-user operating system

12. Why is the UNIX operating system designed with layers?
a. to increase its speed
b. to permit networking
c. to allow other operating systems to communicate with it
d. to insulate the core system from the user environment

13. When viewing a file with the cat command, you can ____________.
a. edit the file
b. only view the file
c. spell check the file
d. delete the file

14. After creating a file with the cat > *filename* command, if you use the command again with the same filename, you ____________.
a. append new information to the file
b. overwrite the file
c. create a new file
d. delete the file

15. The less command fixes the ____________ command's scrolling limitation.
a. cat
b. more
c. clear
d. date

16. UNIX recognizes the first ____________ characters in a user, or login, name.
a. four
b. six
c. eight
d. twelve

17. The –i option used with the who command displays ____________.
a. your login name
b. column headings
c. user idle time information
d. the number of users currently logged in

18. The password should not contain ____________.
a. numbers
b. uppercase letters
c. control characters
d. the underscore character

19. The > symbol may be combined with ______________ to create output.
a. only the cat command
b. any command that produces output
c. only the who command
d. only the cal command

20. The ______________ commands display only the first or last few lines of a file.
a. first and last
b. cat –first and cat –last
c. viewhead and viewtail
d. head and tail

21. Multiple commands entered on the same command line must be separated by a ______________.
a. space
b. semicolon (;)
c. colon (:)
d. comma (,)

22. One major difference between the system administrator and ordinary users is ______________.
a. the system administrator has unlimited permission to alter the system
b. the system administrator does not have to log on to the system
c. the system administrator must enter commands in uppercase letters
d. ordinary users can change some system settings, but not all

23. The symbol that ordinary users see as their command prompt is the ______________.
a. colon (:)
b. pound sign (#)
c. slash character (/)
d. dollar sign ($)

24. True or false: If you forget your password, you can use the passwd command to display it on the screen.

25. True or false: Your password must be different from your user name.

26. Your password must contain at least ______________ character(s).
a. five
b. eight
c. twelve
d. one

27. The passwd command requires that you type your password ______________ times.
a. one
b. two
c. three
d. zero

28. When using the cat > *filename* command, you must enter ________________ to end keyboard input.
a. *
b. Ctrl+D
c. q
d. exit

29. The ________________ symbol directs command output to a file.
a. *
b. $
c. >
d. <

EXERCISES

1. Use the cal command to determine on what day of the week you were born.

2. Use the cal command to discover on what day of the week the Declaration of Independence was signed.

3. Use the cal command to discover what is unusual about the year 1752.

4. Clear the screen and view the online manual for the cal command. What is the explanation for the year 1752?

5. Create a file called today containing today's date.

6. Create a file called manual_for_date containing the online manual for the date command.

7. Use the who command to append a quick list of current users to the date file.

8. View the files today and manual_for_date:
a. individually
b. in sequence

9. View the files today and manual_for_date in sequence by using:
a. the less command
b. the more command

10. Create a file containing the calendar for next year, and use the more command to view the file and scroll forward through it.

11. Use the less command to view the file you created in Exercise 10, and scroll backward through it.

12. Use the tail command to view the last 10 lines of the file you created in Exercise 10.

13. Create a file listing two tasks on your to-do list.

14. Add another task to the list in the file you created in Exercise 13.

DISCOVERY EXERCISES

1. Use the who command to determine how many users are logged on.
2. Use the option for the who command that displays a heading for the information displayed.
3. Combine the cat and cal commands to create a file containing a memo about a meeting to review a new hotel management program. Attach a calendar for November of 2000 to your memo, and remind the recipients of the meeting dates for Wednesday, Thursday, and Friday of the third week of the month.
4. Use the head command to view the first six lines of the file you created in Step 3.
5. Repeat the command you used in Discovery Exercise 4, but redirect the output to another file. What is stored in the file you created?
6. Your system administrator at Dominion mentions that you can also specify the number of bytes you wish to view with the head and tail commands. Use the man program to discover how this is done. Perform the proper commands to view the first and last 10 characters of the file you created in Discovery Exercise 3.